11-8-2021

Matrix Multiplier: Cramer's Method Calculation

Haruka Kido
haruka.kido@und.edu

Matrix Multiplier: Cramer's Method Calculation

Haruka Kido
EE 304: Computer Aided Measurements and Controls
Fall 2020. Final Project

1. Introduction to the Systematic Configuration of 2D Matrix Multiplication

The systemic process of the Cramer's Method enhances the efficiency of solving matrix multiplication problems involving systems of linear equations with sets of 3 or more added or subtracted unknown variables associated with numerical coefficients. The coefficients of the system of equations generate the numbers in the matrix to be used in the replicable and formulaic computation process. As such, turning the Cramer's Method into a computer algorithm and eventually programming the algorithm to a Cramer's Method Calculator Machine with an embedded circuit to read code could be a simple model to optimize engineering computations otherwise done on paper. If done with precision and accuracy to the degree of producing answers to matrix multiplication problems with automation, the configuration between LabVIEW and C++ languages, circuit assemblage, and hardware programming will satisfy the bare foundation to the nature of building a programmed calculator from initialization of scripting to the achievement of algorithmic computations learned by a soldered circuit.

The first code is based on LabVIEW mathematical functions to run array operations for standard Cramer's Method matrix inputs and outputs. The Cramer's Method Calculator program is written in C++ for hardware configurability. The calculator could be used to solve linear systems of equations problems. In LabVIEW, user input permits alteration of matrix size. The C++ code is written for a sample 3*x*3 matrix. The Cramer's Method Calculator codes will be replicable systems to be used in a larger hardware-software project of an external LCD-controller hardware configuration with usability of mathematical symbols and operations based on graphical automations from both or either LabVIEW VIs and C++, for the ultimate capability to be connected to computers to write extended mathematical documents via USB receiver. The electrical engineering hardware component is the extension of this project as Part II. This report and project submission for EE 304 include only Part I, the computer aided part of the Cramer's Method Calculator.

2. The Concept: Cramer's Method Explained

The Cramer's Method involves mathematical steps to solve for a determinant using a 2D matrix with the sets of coefficients of the variables from each linear equation as individual rows. For example, given a system of 3 linear equations, the coefficients associated with the variables distinguish into ordered letter values as shown:

$$\mathbf{4}x + \mathbf{7}y + \mathbf{4}z = 2 \qquad \mathbf{a}x + \mathbf{b}y + \mathbf{c}z = 2$$

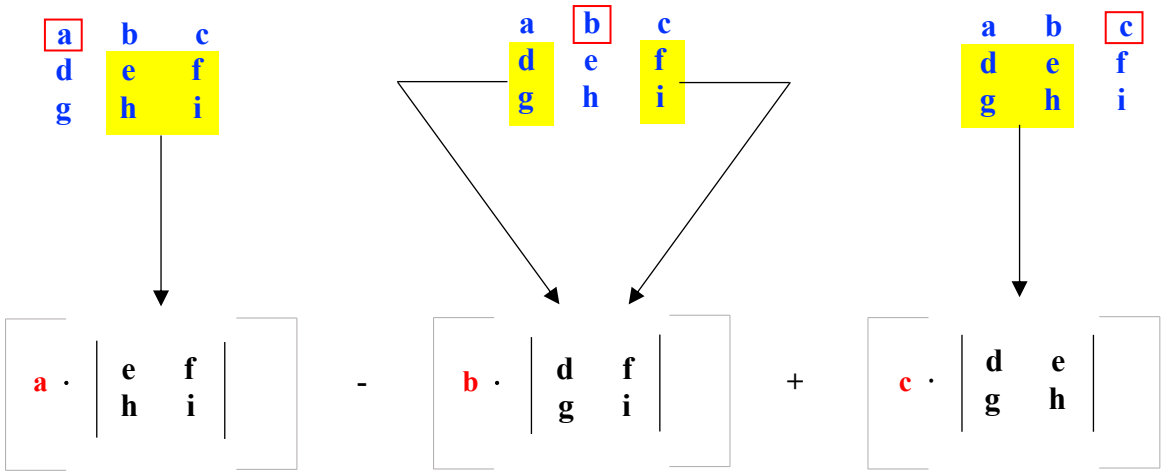
$$\mathbf{8}x + \mathbf{3}y - \mathbf{2}z = 6 \qquad \mathbf{d}x + \mathbf{e}y - \mathbf{f}z = 6$$

$$\mathbf{3}x - \mathbf{6}y + \mathbf{7}z = 3 \qquad \mathbf{g}x - \mathbf{h}y + \mathbf{i}z = 3$$

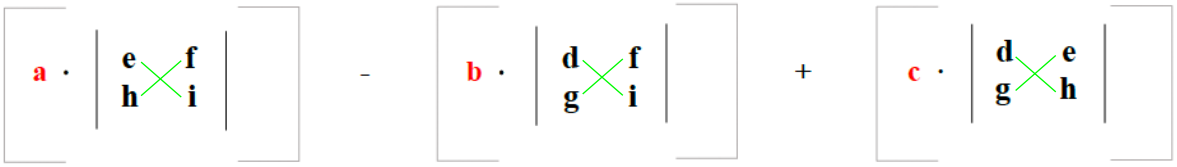
Subsequently, the 3 *x* 3 coefficient matrix of the letter values follows:

$$|M| = \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \\ \mathbf{d} & \mathbf{e} & \mathbf{f} \\ \mathbf{g} & \mathbf{h} & \mathbf{i} \end{bmatrix}$$

To solve for the determinant of the 3 *x* 3 coefficient matrix, the following matrix organization is applied to each original variable (x, y, and z) as 3 matrices. **a** represents x, **b** represents y, and **c** represents z:



Using the organization of each matrix, the values diagonal from each other in each matrix are multiplied:



For each matrix, the first value obtained by the diagonal multiplication (using the upper left and lower right values) subtracts the second value obtained by the diagonal multiplication (using the lower left and upper right values). Each pair of the 2 diagonal values, one subtracted from another, is multiplied by their respective variable, either a, b, or c. The resulting numbers represent the values to be placed in the organization

_____ - ____ + ____ = determinant, as shown below:

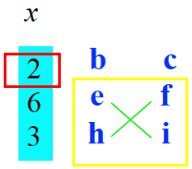
$$\mathbf{D} = a \cdot [(e)(i) - (h)(f)] - b \cdot [(d)(i) - (g)(f)] + c \cdot [(d)(h) - (g)(e)]$$

The above determinant, **D**, is the *denominator* for the 3 fractions which solve for the three variables x, y, and z.

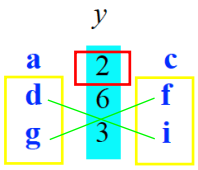
The *numerators* of the 3 fractions are solved for separately. From the original matrix, the column representing the variable associated with that fraction is replaced by the 3 known numbers to the right of the equals signs from the original system of equations. In this case, the 3 numbers are 2, 6, and 3, visualized vertically from top to bottom:

$$\mathbf{N}_x = \begin{matrix} & x & \\ 2 & b & c \\ 6 & e & f \\ 3 & h & i \end{matrix} \qquad \mathbf{N}_y = \begin{matrix} & y & \\ a & 2 & c \\ d & 6 & f \\ g & 3 & i \end{matrix} \qquad \mathbf{N}_z = \begin{matrix} & & z \\ a & b & 2 \\ d & e & 6 \\ g & h & 3 \end{matrix}$$

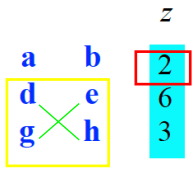
Then, each of the 3 matrices are solved for the *numerators* of the final 3 matrices in the same diagonal matrix multiplication method used for the determinant, except using horizontal row elimination in finding the diagonal pairs to be multiplied by the respective number chosen for that specific variable column, which produces the 3 values to be placed in the organization _____ - ____ + ____ = *numerator*, for each numerator_x, numerator_y, and numerator_z.

$\mathbf{N}_x =$ 

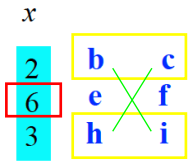
$= 2 \cdot [(e)(i) - (h)(f)]$

$\mathbf{N}_y =$ 

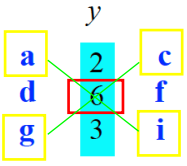
$= 2 \cdot [(d)(i) - (g)(f)]$

$\mathbf{N}_z =$ 

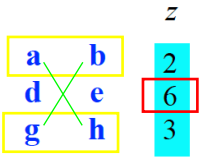
$= 2 \cdot [(d)(h) - (g)(e)]$

$\mathbf{N}_x =$ 

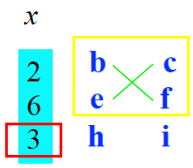
$= 6 \cdot [(b)(i) - (h)(c)]$

$\mathbf{N}_y =$ 

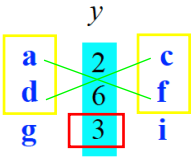
$= 6 \cdot [(a)(i) - (g)(c)]$

$\mathbf{N}_z =$ 

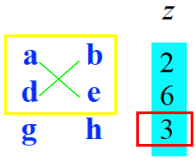
$= 6 \cdot [(a)(h) - (g)(b)]$

$\mathbf{N}_x =$ 

$= 3 \cdot [(b)(f) - (e)(c)]$

$\mathbf{N}_y =$ 

$= 3 \cdot [(a)(f) - (d)(c)]$

$\mathbf{N}_z =$ 

$= 3 \cdot [(a)(c) - (d)(b)]$

$\mathbf{N}_x =$

$2 \cdot [(e)(i) - (h)(f)]$

-

$6 \cdot [(b)(i) - (h)(c)]$

+

$3 \cdot [(b)(f) - (e)(c)]$

$\mathbf{N}_y =$

$2 \cdot [(d)(i) - (g)(f)]$

-

$6 \cdot [(a)(i) - (g)(c)]$

+

$3 \cdot [(a)(f) - (d)(c)]$

$\mathbf{N}_z =$

$2 \cdot [(d)(h) - (g)(e)]$

-

$6 \cdot [(a)(h) - (g)(b)]$

+

$3 \cdot [(a)(c) - (d)(b)]$

Therefore, the solutions ("Solution Vector") for the 3 variables using the Cramer's Method include:

$$x = (N_x) / (D) \qquad y = (N_y) / (D) \qquad z = (N_z) / (D)$$

3. Building the Cramer's Method Algorithm into LabVIEW:

Using the conceptual methodology of solving for the unknown variables given a system of linear equations, the graphical programming language LabVIEW is used to simulate a block diagram to run a problem requiring matrix multiplication through the Cramer's Method. The LabVIEW VI includes the ability for user inputs of coefficients into a matrix of any size. Given the coefficient inputs and matrix size selection, the VI schematic outputs the solution to the unknown variables. The LabVIEW program shows 3*x*3 as the sample matrix size. The 3 known values to the right of the equals signs of the linear equations are considered as the inputs of the "Known Vector." The "Input Matrix" likewise requires numerical input for the solution variables to be generated as the "Solution Vector." This process underlying the components of the Cramer's Method is built from the Mathematics Function called "Solve Linear Equations," which configures to an embedded .lvlib.vi. The front panel should have the same values and system assemblage as the embedded lvlib.vi from the matrix multiplier originating in the block diagram.

Controls: Input Matrix (as coefficients of the variables x, y, z for a 3*x*3 matrix), and Known Vector

Mathematical Function: Linear Algebra (represents the Cramer's Calculator explained above)

Indicator: Solution Vector

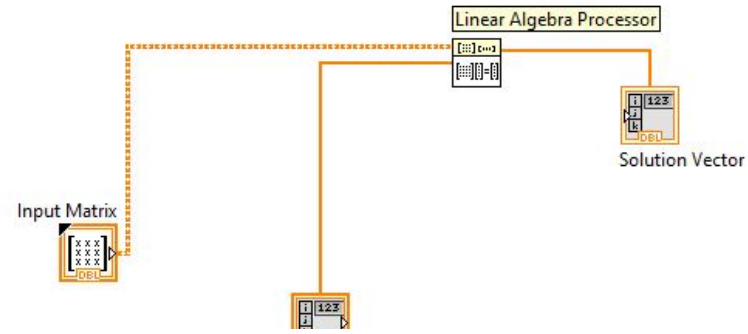


Figure 1. Block Diagram

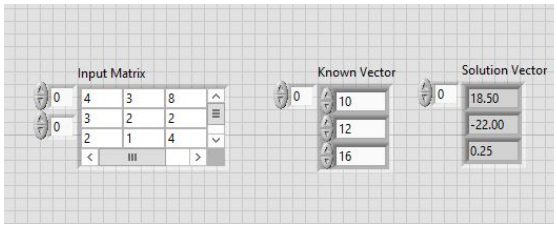


Figure 2. Front Panel

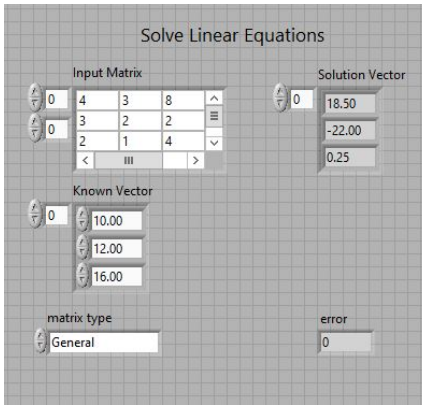


Figure 3. Linear Equations Input and Output Values

To confirm the accuracy of the block diagram to calculate the solutions to a matrix multiplication problem with the Cramer's Method, the raw calculations are shown below:

$$4x + 3y + 8z = 10$$

$$3x + 2y + 2z = 12$$

$$2x + 1y + 4z = 16$$

In solving for the determinant as the denominator of the 3 fractions:

$$D = (4)[(2)(4) - (2)(1)] - (3)[(3)(4) - (2)(2)] + (8)[(3)(1) - (2)(2)]$$

$$= (4)[8 - 2] - (3)[12 - 4] + (8)[3 - 4]$$

$$= (4)[6] - (3)[8] + (8)[-1]$$

$$= (24) - (24) + (-8)$$

$$= -8$$

In solving for the numerators of the 3 fractions:

$\mathbf{N_x} = (10)\,[(2)(4) - (2)(1)] \;-\; (12)\,[(3)(4) - (8)(1)] \;+\; (16)[(3)(2) - (2)(8)]$

$= (10)[8 - 2] \;-\; (12)[12 - 8] \;+\; (16)[6 - 16] \;=\; (10)[6] - (12)[4] + (16)[-10] \;=\; (60) - (48) + (-160) \;=\; -148$

$\mathbf{N_y} = (10)[(3)(4) - (2)(2)] \;-\; (12)[(4)(4) - (8)(2)] \;+\; (16)[(4)(2) - (8)(3)]$

$= (10)[12 - 4] \;-\; (12)[16 - 16] \;+\; (16)[8 - 24] \;=\; (10)[8] - (12)[0] \;+\; (16)[-16] \;=\; (80) - (0) + (-256) \;=\; -176$

$\mathbf{N_z} = (10)[(3)(1) - (2)(2)] \;-\; (12)[(4)(1) - (3)(2)] \;+\; (16)[(4)(2) - (3)(3)]$

$= (10)[3 - 4] \;-\; (12)[4 - 6] \;+\; (16)[8 - 9] \;=\; (10)[-1] \;-\; (12)[-2] \;+\; (16)[-1] = (-10) - (-24) + (-16) \;=\; -2$

To resolve sign convention issues, it is observed that the mathematical function in LabVIEW follows the commutative property for the subtraction between diagonal multiplication values. The Solution Vector outputs are indeed accurate. *Note that LabVIEW used the commutative property for the y value.

$\mathbf{N_x} / \mathbf{D} = (-148) / (-8) = \mathbf{18.50}$ $\mathbf{N_y} / \mathbf{D} = (-176) / (-8) = \mathbf{22}$ $\mathbf{N_z} / \mathbf{D} = (-2) / (-8) = \mathbf{.25}$

4. Scripting the Cramer's Method into C++ for Preliminary of Part II, Electrical Engineering of Algorithmic Processor:

The C++ code asks for the user to input the coefficient values for the 3*x*3 matrix, including the values for the solution vector. The program then prints out the linear system of equations for the matrix, as per the user's inputs. Subsequently, the calculations follow the Cramer's Method rules to calculate the determinant and the numerators for the solution fractions. The algorithm replaces the column for the associated variable, either X, Y, or Z, with the solution vector column (regarded as the input values for d1, d2, and d3), to calculate the fraction numerators for each variable. Furthermore, the program prints the results of the determinant, the numerators, the X =, Y =, and Z = fractions, and the solutions to X, Y, and Z. The sample input values are congruent with the LabVIEW sample values and the values used for the calculations done without computer algorithms.

```cpp
#include <iostream>
#include <windows.h>
#include <iostream>
#include <iomanip>
#include <stdio.h>

using namespace std;

int main(){
    double Deter, DeterX, DeterY, DeterZ, A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, D3;
    double DeterX_Dem, DeterY_Dem, DeterZ_Dem;
    bool Det_Ind=false;

    cout<<"Cramer's Method Calculation"<<endl;
    cout<<"Enter a1: ";
    cin>>A1;
    cout<<"Enter b1: ";
    cin>>B1;
    cout<<"Enter c1: ";
    cin>>C1;
    cout<<"Enter d1: ";
    cin>>D1;
    cout<<"Enter a2: ";
    cin>>A2;
    cout<<"Enter b2: ";
    cin>>B2;
    cout<<"Enter c2: ";
    cin>>C2;
    cout<<"Enter d2: ";
    cin>>D2;
    cout<<"Enter a3: ";
    cin>>A3;
    cout<<"Enter b3: ";
    cin>>B3;
    cout<<"Enter c3: ";
    cin>>C3;
    cout<<"Enter d3: ";
    cin>>D3;

    cout<<"     "<<endl;
    cout<<"The Linear System of Equations for the Matrix is: "<<endl;
    cout<<A1<<"x + "<<B1<<"y + "<<C1<<"z = "<<D1<<endl;
    cout<<A2<<"x + "<<B2<<"y + "<<C2<<"z = "<<D2<<endl;
    cout<<A3<<"x + "<<B3<<"y + "<<C3<<"z = "<<D3<<endl;

    Deter=(A1*(B2*C3)-A1*(B3*C2)) - (B1*(A2*C3)-B1*(A3*C2)) + (C1*(A2*B3)-C1*(A3*B2));
    DeterX=(D1*(B2*C3)-D1*(B3*C2)) - (D2*(B1*C3)-D2*(B3*C1)) + (D3*(B1*C2)-D3*(B2*C1));
    DeterY=(D1*(A2*C3)-D1*(A3*C2)) - (D2*(A1*C3)-D2*(A3*C1)) + (D3*(A1*C2)-D3*(A2*C1));
    DeterZ=(D1*(A2*B3)-D1*(A3*B2)) - (D2*(A1*B3)-D2*(A3*B1)) + (D3*(A1*B2)-D3*(A2*B1));

    if (Deter==0 && DeterX==0 && DeterY==0 && DeterZ==0) {
                cout<<"Infinite Solutions"<< endl;
                Det_Ind = true;
                }
    else if (Deter==0 && (DeterX!=0 || DeterY!=0 || DeterZ != 0)){
                cout<<"System Error"<< endl;
                Det_Ind = true;
                }

    cout<<"     "<<endl;
    cout<<"D    = "<< Deter << endl;
    cout<<"D(x) = "<< DeterX << endl;
    cout<<"D(y) = "<< DeterY << endl;
    cout<<"D(z) = "<< DeterZ << endl;
    cout<<"     "<<endl;

    if (Det_Ind==true){
                cout<<"Press Enter to continue.";
                cin.get();
                cin.get();
                cin.get();
                return 0;
                }

    else if (Det_Ind==false){
            DeterX_Dem=(DeterX/Deter);
            DeterY_Dem=(DeterY/Deter);
            DeterZ_Dem=(DeterZ)/(Deter);
            cout<<"The Solution Fractions Include: "<<endl;
            cout<<"X = "<<DeterX<<"/"<<Deter<<" = "<<DeterX_Dem<<endl;
            cout<<"Y = "<<DeterY<<"/"<<Deter<<" = "<<DeterY_Dem<<endl;
            cout<<"Z = "<<DeterZ<<"/"<<Deter<<" = "<<DeterZ_Dem<<endl;
            cout<<"     "<<endl;

            cout<<"Solution Set is {("<<DeterX_Dem<<", "<<DeterY_Dem<<", "<<DeterZ_Dem<<")}"<<endl;
            }

    cout<<"Press Enter to continue.";
    cin.get();
    cin.get();
    cin.get();
    return 0;
}
```

```
C:\Users\Haruka Kido\Desktop\3x3 Cramer's Method.exe
Enter a1: 4
Enter b1: 3
Enter c1: 8
Enter d1: 10
Enter a2: 3
Enter b2: 2
Enter c2: 2
Enter d2: 12
Enter a3: 2
Enter b3: 1
Enter c3: 4
Enter d3: 16

The Linear System of Equations for the Matrix is:
4x + 3y + 8z = 10
3x + 2y + 2z = 12
2x + 1y + 4z = 16

D    = -8
D(x) = -148
D(y) = -176
D(z) = -2

The Solution Fractions Include:
X = -148/-8 = 18.5
Y = -176/-8 = 22
Z = -2/-8 = 0.25

Solution Set is {(18.5, 22, 0.25)}
Press Enter to continue.
```